Raspberry Pi MAX7219 Driver Documentation

Release 0.2.3

Richard Hull

Jun 17, 2017

Contents

Interfacing LED matrix displays with the MAX7219 driver [PDF datasheet] in Python (both 2.7 and 3.x are supported) using hardware SPI on the Raspberry Pi. A LED matrix can be acquired for a few pounds from outlets like Banggood. Likewise 7-segment displays are available from Ali-Express or Ebay.

This library supports:

- multiple cascaded devices
- LED matrix and seven-segment variants

Python Usage

For the matrix device, initialize the `matrix` class:

```python
import max7219.led as led

device = led.matrix()
device.show_message("Hello world!")
```

For the 7-segment device, initialize the `sevensegment` class:

```python
import max7219.led as led

device = led.sevensegment()
device.write_number(deviceId=0, value=3.14159)
```

The MAX7219 chipset supports a serial 16-bit register/data buffer which is clocked in on pin DIN every time the clock edge falls, and clocked out on DOUT 16.5 clock cycles later. This allows multiple devices to be chained together.

When initializing cascaded devices, it is necessary to specify a `cascaded=...` parameter, and generally methods which target specific devices will expect a `deviceId=...` parameter, counting from zero.

For more information, see https://max7219.readthedocs.io/

Pre-requisites

By default, the SPI kernel driver is **NOT** enabled on the Raspberry Pi Raspian image. You can confirm whether it is enabled using the shell commands below:

```
$ lsmod | grep -i spi
spi_bcm2835              7424  0
```

Depending on the kernel version, this may report **spi_bcm2807** rather than **spi_bcm2835** - either should be adequate.

And that the devices are successfully installed in `/dev`:

```
$ ls -l /dev/spi*
crw------- 1 root root 153, 0 Jan  1  1970 /dev/spidev0.0
crw------- 1 root root 153, 1 Jan  1  1970 /dev/spidev0.1
```

If you have no `/dev/spi` files and nothing is showing using `lsmod` then this implies the kernel SPI driver is not loaded. Enable the SPI as follows (steps taken from https://learn.sparkfun.com/tutorials/raspberry-pi-spi-and-i2c-tutorial#spi-on-pi):

1. Run `sudo raspi-config`

2. Use the down arrow to select `9 Advanced Options`

3. Arrow down to `A6 SPI.`

4. Select **yes** when it asks you to enable SPI,

5. Also select **yes** when it asks about automatically loading the kernel module.

6. Use the right arrow to select the **<Finish>** button.

7. Select **yes** when it asks to reboot.

After rebooting re-check that the `lsmod | grep -i spi` command shows whether SPI driver is loaded before proceeding. If you are stil experiencing problems, refer to the official Raspberry Pi SPI troubleshooting guide for further details, or ask a new question - but please remember to add as much detail as possible.

GPIO pin-outs

The breakout board has two headers to allow daisy-chaining:

Board Pin	Name	Remarks	RPi Pin	RPi Function
1	VCC	+5V Power	2	5V0
2	GND	Ground	6	GND
3	DIN	Data In	19	GPIO 10 (MOSI)
4	CS	Chip Select	24	GPIO 8 (SPI CE0)
5	CLK	Clock	23	GPIO 11 (SPI CLK)

Note: See below for cascading/daisy-chaining, power supply and level-shifting.

Installing the library

Note: The library has been tested against Python 2.7 and 3.4. For **Python3** installation, substitute `pip` `pip3`, `python` `python3`, `python-dev` `python3-dev`, and `python-pip` `python3-pip` in the instructions below.

Install the latest version of the library directly from PyPI:

```
$ sudo apt-get install python-dev python-pip
$ sudo pip install max7219
```

Alternatively, clone the code from github:

```
$ git clone https://github.com/rm-hull/max7219.git
$ cd max7219
$ sudo pip install -e .
```

Next, follow the specific steps below for your OS.

Raspbian

```
$ cd max7219
$ sudo apt-get install python-dev python-pip
$ sudo pip install spidev
$ sudo python setup.py install
```

Arch Linux

```
cd max7219
pacman -Sy base-devel python2
pip install spidev
python2 setup.py install
```

Cascading, power supply & level shifting

The MAX7219 chip supports cascading devices by connecting the DIN of one chip to the DOUT of another chip. For a long time I was puzzled as to why this didnt seem to work properly for me, despite spending a lot of time investigating and always assuming it was a bug in code.

- Because the Raspberry PI can only supply a limited amount of power from the 5V rail, it is recommended that any LED matrices are powered separately by a 5V supply, and grounded with the Raspberry PI. It is possible to power one or two LED matrices directly from a Raspberry PI, but any more is likely to cause intermittent faults & crashes.

- Also because the GPIO ports used for SPI are 3.3V, a simple level shifter (as per the diagram below) should be employed on the DIN, CS and CLK inputs to boost the levels to 5V. Again it is possible to drive them directly by the 3.3V GPIO pins, it is just outside tolerance, and will result in intermittent issues.

Despite the above two points, I still had no success getting cascaded matrices to work properly. Revisiting the wiring, I had connected the devices in serial connecting the out pins of one device to the in pins of another. This just produced garbled images.

Connecting the CLK lines on the input side all together worked first time. I can only assume that there is some noise on the clock line, or a dry solder joint somewhere.

If you have more than one device and they are daisy-chained together, you can initialize the library with:

```python
import max7219.led as led

device = led.matrix(cascaded = 3)
device.show_message("Hello world!")
```

To address a specific device, most other methods expect a `deviceId=N` parameter (where N=0..cascaded-1).

Examples

Run the example code as follows:

```
$ sudo python examples/matrix_test.py
```

or:

```
$ sudo python examples/sevensegment_test.py
```

Note: By default, SPI is only accessible by root (hence using sudo above). Follow these instructions to create an spi group, and adding your user to that group, so you don't have to run as root.

CHAPTER 7

References

- http://hackaday.com/2013/01/06/hardware-spi-with-python-on-a-raspberry-pi/
- http://gammon.com.au/forum/?id=11516
- http://louisthiery.com/spi-python-hardware-spi-for-raspi/
- http://www.brianhensley.net/2012/07/getting-spi-working-on-raspberry-pi.html
- http://raspi.tv/2013/8-x-8-led-array-driven-by-max7219-on-the-raspberry-pi-via-python
- http://quick2wire.com/non-root-access-to-spi-on-the-pi

Contributing

Pull requests (code changes / documentation / typos / feature requests / setup) are gladly accepted. If you are intending some large-scale changes, please get in touch first to make sure we're on the same page: try and include a docstring for any new methods, and try and keep method bodies small, readable and PEP8-compliant.

Contributors

- Thijs Triemstra (@thijstriemstra)
- Jon Carlos (@webmonger)
- Unattributed (@wkapga)
- Taras (@tarasius)
- Brice Parent (@agripo)

License

CHAPTER 10

Indices and tables

- genindex
- modindex
- search